Computer Networks and Communications
https://ojs.wiserpub.com/index.php/CNC

UNIVERSAL WISER PUBLISHER

Research Article

Estimating Throughput in Optical Backbone Networks Using Deep Neural Networks

Alexandre Freitas[1], João Pires[2*]

[1] Department of Electrical and Computer Engineering, Instituto Superior Técnico, Universidade de Lisboa, Lisboa, Portugal
[2] Department of Electrical and Computer Engineering and Instituto de Telecomunicações, Instituto Superior Técnico, Universidade de Lisboa, Lisboa, Portugal
E-mail: jpires@lx.it.pt

Received: 1 November 2024; **Revised:** 19 December 2024; **Accepted:** 6 January 2025

Abstract: Optical backbone networks, which use optical fibres as the transmission medium, form the core infrastructure used by network operators to deliver services to users, as well as by Internet companies to route traffic between data centres. The network throughput is a key parameter in the analysis of the networks' performance. However, its determination can be a complex process that involves long computation times, since aspects related to both the physical and network layers need to be accounted for. To face this challenge, we propose a machine learning solution: a deep neural network (DNN) model, that has the goal of estimating the values of the network throughput and of a closely related parameter, average channel capacity, accurately and with short computation times. The simulation results indicate that the DNN model accurately predicts both outputs, with mean relative errors of 6.17% for the network throughput and 2.84% for the average channel capacity. These predictions are made in just a few milliseconds, providing a significant advantage over the heuristic routing algorithms, which can take up to tens of seconds in larger networks.

Keywords**:** optical network, network throughput, deep neural network, machine learning

1. Introduction

Recent years have seen a sustained increase in data traffic, a trend expected to persist as bandwidth-intensive services continue to expand. Key drivers of this growth include video streaming, social media, cloud computing, and advancements in technologies such as 5G and artificial intelligence applications. This growing traffic places considerable pressure on the networks of telecommunications operators, particularly in the backbone segments, making the design of future optical networks, which are the fundamental infrastructure for delivering these services, a significant challenge [1, 2].

Optical networks are communication infrastructures, typically owned by telecommunication operators (telcos) or Internet companies, that use light signals transmitted through optical fibres to efficiently transfer data over long distances at huge bit rates. The enormous bandwidth of optical fibres enables the simultaneous transmission of multiple optical signals (also referred to as optical channels), each operating at a unique wavelength, over a single fibre. This technique is called wavelength division multiplexing (WDM). The fibre bandwidth is divided into optical bands according to established standards, with the C-band being the most widely used. The number of optical channels that can be carried over an optical fibre is typically limited to about one hundred when utilizing the named extended C-band, which has a bandwidth of

DOI: https://doi.org/10.37256/cnc.3120256008

around 4800 GHz. This limitation may cause traffic blocking if no additional channels are accessible to route new traffic demands. This problem can be overcome by employing other optical bands beyond the C-band, as well as space-division multiplexing techniques, such as using multiple optical fibre pairs per link rather than just one, as is typical [3].

Network throughput is a key performance metric for optical networks, describing the rate at which data is successfully transmitted. It is equivalent to network capacity in the absence of traffic blocking. This capacity is defined as the maximum amount of data the network can transfer across all its nodes per unit of time and is closely tied to the concept of channel capacity, as introduced by Claude Shannon in 1948 [4].

To estimate the optical network capacity, it is necessary not only to determine the optical channel capacity, which is related to physical layer aspects including the characteristics of optical fibers, but also to consider topological aspects, traffic demands, routing, as well as wavelength and modulation assignment.

The estimating of optical channel capacity has been the focus of many studies, with some relying on accurate numerical simulations [5] and others on analytical models [6, 7].

Similarly, the topic of optical network capacity and network throughput have also received considerable attention. In [8], the authors presented an algorithm to maximize the througput of an optical network in the presence of physical layer impairments. The algorithm was based on an integer linear program (ILP) and aimed to optimize routing, wavelength assignment, and other system parameters. An alternative approach to capacity estimation, which used a heuristic algorithm for routing and wavelength assignment instead of the ILP was presented in [9]. To study the impact of network topology on network throughput, [10] introduced a new graph model based on the Barabási-Albert model. This modified version includes physical layer aspects and has been shown to outperform classical models in maximizing network throughput. Additionally, a framework was proposed to explore the relationship between various topological parameters and network performance metrics, including network capacity [11]. This framework offered important insights into the critical factors that affect network capacity. Our recent work, [12], also addresses the issue of optical backbone network capacity. In this study, we evaluated the impact of topological and physical aspects on network capacity. For this purpose, we generated a set of random graphs that accurately describe real optical backbone networks and developed appropriate routing strategies. This framework is also used in the present work as a simulator, enabling the generating of adequate datasets.

The estimation of optical backbone network capacity, even when using analytical methods to deal with channel capacity, faces the challenge of long computation times, particularly for large-scale networks [13]. Given the computational advantage of machine learning (ML) techniques in solving complex problems, it makes sense to explore their utilization to address this issue. These techniques have been used extensively in the context of optical networking (see [14] and references therein). Among different ML techniques, neural networks, and more specifically deep neural networks (DNNs), which are neural networks with multiple hidden layers of units (neurons) between the input and output layers, have been proven to achieve better results in various contexts [15, 16, 17]. In particular, [15] demonstrated that DNNs exhibit superior generalization capabilities compared to traditional ML models, such as XGBoost and others. Additionally, [16] showed that DNNs achieve higher prediction accuracy in optical fibre design compared to XGBoost and random forests (RF). Similarly, [17] highlighted the advantages of NNs in optical network design over RF and logistic regression in regression operations. Furthermore, to deal with regression problems, such as network capacity estimation, it has been shown that neural networks with at least one hidden layer can approximate any continuous function with any desired degree of accuracy, provided they have enough neurons [18]. This makes them well-suited for problems characterized by particularly complex data relationships.

DNNs have been the focus of many applications in the field of optical networks, including quality of transmission prediction [19], optical signal-to-noise ratio monitoring [20], and routing and spectrum assignment [21]. However, to the best of the authors' knowledge, they have not yet been applied specifically to network throughput studies.

In this paper, we propose the utilization of DNN techniques to address the problem of estimating the throughput of optical backbone networks, as well as the average channel capacity, assuming a uniform traffic demand pattern between all network nodes. To train the DNN model, a dataset of 15,245 random networks was generated using the tool described in [12]. The estimation capabilities of the model are very good for network sizes included in the training dataset, but it finds it difficult to generalize and provide accurate results for network sizes outside this range.

The remainder of this paper is organized as follows: Section 2 provides an overview of key aspects of network modelling and random network generation. Section 3 introduces the proposed DNN model. Section 4 discusses simulation results and highlights the model's accuracy. Finally, Section 5 concludes the paper.

2. Networking aspects

2.1 *Network modelling*

In our analysis, we assume that the optical backbone networks are transparent, meaning that optical signals are transmitted from the source to the destination without conversion to electrical signals, maintaining their optical nature throughout the entire network. Therefore, in these networks, all node functionalities (such as multiplexing, switching, routing, etc.) take place in the optical domain, and the node structure is based on reconfigurable optical add-drop multiplexers (ROADMs). The ROADM is responsible not only for locally adding and dropping optical channels but also for performing optical bypassing by switching optical channels from the incoming to the outgoing optical links. In terms of switching paradigms, an optical network can be seen as a circuit-switched network, where the circuits correspond to optical channels and ROADMs serve as circuit switches. An optical link is a physical interconnection between two adjacent nodes, consisting of a pair of optical fibres to support bidirectional traffic. The link also includes optical amplifiers strategically placed to compensate for fibre losses. Each optical fibre carries WDM signals, comprising multiple optical channels, with each channel defined by its unique wavelength. The reliance on wavelengths set optical networks apart from electrical ones, making the wavelength assignment a key design problem in the former. The count of optical channels in a single link, termed as N_{ch}, is determined by the channel frequency spacing, which in this work is assumed be equal to the baud rate R_s, and the bandwidth of the optical amplifier, which we consider to be equal to the bandwidth of the C-band (4800 GHz).

An optical network can be modelled as an undirected weighted graph $G(V,E)$, where $V = \{v_1, \ldots, v_N\}$ is a set of nodes and $E = \{e_1, \ldots, e_K\}$ is a set of links, with $N = |V|$ being the number of nodes, and $K = |E|$ being the number of links. Each link $(v_i, v_j) \in E$ is characterized by two attributes: (i) $l_{i,j}$ the length of the link in kilometres between the nodes v_i and v_j; (ii) $u_{i,j}$, the link capacity, expressed as the number of channels N_{ch}.

Besides N and K, other important parameters of a graph G include the node degree $\delta(G)$, the network diameter $d(G)$, and the algebraic connectivity $a(G)$. $\delta(G)$ indicates the number of links connected to a given node, $d(G)$ represents the length of the longest shortest path between any two nodes, while $a(G)$ is the second smallest eigenvalue of the Laplacian matrix L, which is a $N \times N$ matrix whose elements are given as [22]

$$L_{i,j} = \begin{cases} \delta(v_i) & \text{if } i = j \\ -1 & \text{if } v_i \text{ is adjacent to } v_j, \mathrm{i} \neq \mathrm{j} \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where $\delta(v_i)$ denotes the degree of node v_i. Algebraic connectivity provides valuable structural information about network connectivity and robustness and, in this way, complements other metrics like degree and diameter. Furthermore, as shown in [11] the algebraic connectivity is strongly correlated with the network's capacity.

In the context of DNNs, large datasets are required for training and testing the model. Due to the unavailability of network monitoring data or experimental data, we rely on synthetic datasets. To address this, we generated a large number of network topologies using random graphs designed to accurately represent the characteristics of real-world optical networks. In [12], we presented a tool we developed for generating random networks suitable for modelling optical backbone networks. The tool is based on a modified Waxman model and is capable of generating networks resilient to single-link failures. This model works by dividing a two-dimensional (2D) square plane with area $A = L^2$, where L is the side length of the plane, into a set of regions. In the first step of the process, N nodes are randomly placed within these regions. In the subsequent steps, nodes are interconnected with links both within each region and across different regions according to the Waxman probability, subject to certain constrains on node degree and λ-connectivity, where λ represents the maximum number of link disjoint paths between two nodes. The Waxman probability is characterized by the parameters α and β , both within the range [0,1], with α influencing the link length and β controlling the link density [23].

2.2 *Traffic, RWA and network throughput modelling*

Network throughput is closely related with network capacity, which is defined as the maximum amount of data that can be transferred across all network nodes in a given period of time. This capacity depends on various network properties, such as the physical and logical topology, physical parameters, link capacity, node structure, routing, and wavelength assignment, etc. The physical topology defines the interconnection pattern of nodes and is represented by the graph $G(V,E)$. A key step in evaluating network capacity is defining the logical topology, which describes the profile of traffic demands between nodes. This profile is defined by the demand matrix $D = [d_{s,t}]$, which is a $N \times N$ matrix, where each entry $d_{s,t}$ represents the number of traffic demands between source node s and termination node t with $s,t \in V$. In this analysis, we assume a uniform demand profile of $d_{s,t} = 1$ for all node pairs, corresponding to a full-mesh logical topology in which each node is logically connected to every other node in the network.

For each traffic demand $d_{s,t}$, it is necessary to find a lightpath $\pi_{s,t}$ in the physical topology between each pair of nodes and assign a wavelength to this path. This operation is known as routing and wavelength assignment (RWA). Given that there are multiple paths between each pair of nodes, the routing objective is to determine the shortest path using a heuristic such as Dijkstra's algorithm. The shortest path is the one that minimizes the total path length, defined as the sum of the lengths of all the links traversed by the path, represented as $l\left(\pi_{s,t}\right) = \sum_{i,j} l_{i,j}$. Furthermore, the wavelength assignment must ensure that the same wavelength is used for all the links traversed by the lightpath to maintain continuity, and that no two lightpaths can share the same wavelength on a single link. Each lightpath is physically established using an optical channel characterized by its wavelength λ_k and capacity $C_{ch,k}$, specifically for the case of channel $k = (s,t)$. Note that $k \in S$, where $S = \{1,\ 2,\dots,N(N-1)\}$ is the set of channels required to implement a logical full mesh topology.

By knowing the path length $l\left(\pi_k\right)$, it is possible to compute the maximum capacity $C_{ch,k}$ of the corresponding optical channel, also known as the Shannon capacity, given in bits per second. This calculation makes use of the optical reach values of the lightpath (see Table 2 of [12]), where optical reach is defined as the maximum path length that can achieve a specified capacity, assuming a baud rate of 64 Gbaud.

Having in mind that the total bandwidth available per link is 4800 GHz, we arrive at the maximum of 75 optical channels per link ($N_{ch.max} = 75$) for a baud rate of 64 Gbaud. This limitation in link capacity can result in traffic demand blocking when no free optical channel is available on at least one of the links traversed by the lightpath. By applying a constrained routing algorithm along with a first-fit wavelength assignment [12, 24] (see Algorithm 1 in the Appendix), we obtain the matrix of established paths, $\Pi = [\pi_k]$, and the matrix of blocked paths, $B = [b_k]$, where

$$b_k = \begin{cases} 1 & \text{if path } \ \pi_k \text{ is blocked} \\ 0 & \qquad \text{otherwise.} \end{cases} \tag{2}$$

This allows us to determine the average blocking ratio, given by

$$\overline{B}_R = \sum_{k\in S} b_k / N(N-1). \tag{3}$$

According to the previous discussion, network capacity can be redefined as the maximum amount of data that the entire network can transfer across all optical channels simultaneously, per unit of time. It can therefore be calculated by summing the capacities of all optical channels in the set S, giving

$$C_{net} = \sum_{k\in S} C_{ch,k}. \tag{4}$$

Another important metric in our study is the network throughput. This metric describes the actual data rate achieved across the entire networks, but taking into account real operating conditions, such as the presence of blocking. Consequently, we define network throughput in the following way:

$$T_{net} = C_{net}\left(1 - \overline{B}_R\right). \tag{5}$$

An additional metric for network analysis is the network-wide average channel capacity, defined as [25]

$$\overline{C}_{ch} = \frac{\sum_{k \in S} C_{ch,k}}{\sum_{k \in S} \gamma_k} \tag{6}$$

where γ_k denotes the expected utilization ratio of channel k. For the sake of simplicity, it is assumed that $\gamma_k = 1$ for all channels. As a result, the sum in the denominator of (6) equals the total number of paths in the network, which for a full-mesh logical topology, amounts to $N(N-1)$. With this simplification, the average channel capacity reduces to

$$\overline{C}_{ch} = C_{net}/N(N-1). \tag{7}$$

3. Neural network design

3.1 *Model*

A four-layer neural network is used to implement the model used in this study. The first layer is the input layer, the second and third layers are hidden layers, and the fourth layer is the output layer. A schematic diagram of this model is depicted in Figure 1. Mathematically, the set of input features is described by the vector $X = [x_1,\ x_2 \ldots\ldots x_n]$ and the set of outputs by the vector $Y = [y_1,\ y_2]$ where y_1 and y_2 denote the predicted value for the network throughput (T_{net}) and average channel capacity ($\overline{C}_{ch}$), respectively. Furthermore, to characterize the model it is necessary do define the weight matrices:

- W_1, between the input layer and the first hidden layer with size $(n,\ m_1)$,
- W_2, between the first hidden layer and second hidden layer with size (m_1, m_2),
- W_3, between the second layer and the output layer with size $(m_2, 2)$.

As well as the bias vectors:

- b_1, for the first hidden layer with size (1, m_1),
- b_2, the second hidden layer with size (m_1, m_2),
- b_3, for the output layer with size (1,2).

To learn complex patterns in data, the input-output relationships in the hidden layers of a DNN must be non-linear. In this work, this non-linearity is achieved through activation functions such as the ReLU (Rectified Linear Unit) function, defined as

$$g(x) = \max(0,\ x). \tag{8}$$

For the output layer, a linear activation function is instead used, given by

$$f(x) = x. \tag{9}$$

Note that both these activation functions are typically used in regression problems [14, 26].

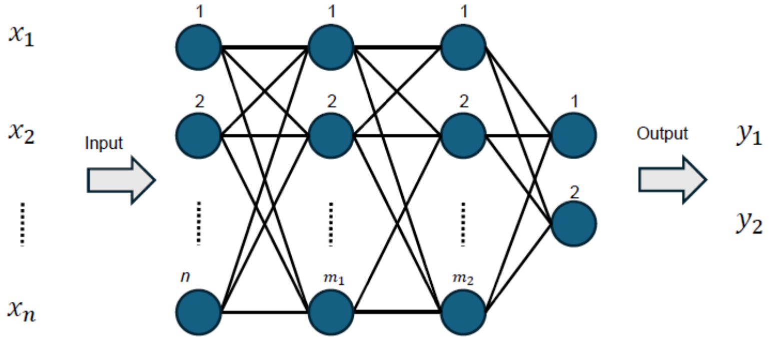

Figure 1. Model of a neural network with 2 hidden layers

3.2 *Training neural networks*

The training of neural networks involves determining the values of all W matrices and bias vectors that minimize a given loss function using an appropriate iterative method (optimizer algorithm).

In the training process, a dataset consisting of features (vector X) and labels (vector Y) is split into three subsets: a training set, a validation set, and a test set. The training set is used by the model to learn by adjusting its internal parameters (weights and biases) in order to minimize a loss function. The validation set permits an unbiased evaluation of the model's fit on the training data while tuning various model parameters, also known as hyperparameters. Finally, the test set is used to evaluate the performance of the optimized trained model on the previously unseen data. Before the data is split into these three sets, it needs to be pre-processed and shuffled. Data pre-processing consists in preparing the data to make it more suitable for the training process.

The loss function measures the difference between the value predicted by the neural network (NN) and the actual value. In other words, it quantifies the error in the model's predictions. For regression problems, the Mean Squared Error (MSE) is normally used [26]. The MSE can be written as

$$MSE = \frac{1}{M}\sum_{i=1}^{M}(\widehat{y}_i - y_i)^2 \tag{10}$$

where M is the number of data values being considered, $\widehat{y}_i$ are the estimated values and y_i are the actual values.

The optimizer algorithm determines how the weight matrices and bias vectors are adjusted during training. Common optimizers include the Stochastic Gradient Descent (SGD) and the Adaptive Momentum Estimation (Adam). Updating the network parameters requires computing the gradient of the loss function, a task performed by the backpropagation algorithm [27]. The backpropagation algorithm is used in neural networks to calculate the gradient of the loss function with respect to the network's weights and biases by applying the chain rule of calculus and propagating the loss backward from the output layer to the input layer. The resulting gradients, which indicate the direction and rate of change of the loss function, guide the optimizer algorithm in iteratively adjusting the weights and biases. An important parameter of the optimizer is the learning rate, which determines the magnitude of the updates applied to the weights and biases during each iteration.

Optimizing the hyperparameters is an essential part of training a NN. Hyperparameters are the variables that control how the model learns from the data. These include the number of hidden layers, the number of hidden units, and the learning rate. During the training, different combinations of hyperparameters are tested to find the one that results in the best performance on the validation set.

In this work, the performance of the model on the validation set is assessed using two metrics: the R^2 score and the Mean Relative Error (*MRE*). The first metric is defined as

$$R^2 = 1 - \frac{\sum_{i=0}^{M} (y_i - \widehat{y}_i)^2}{\sum_{i=0}^{M} (y_i - \overline{y}_i)^2} \tag{11}$$

where y_i represents the actual value, $\widehat{y}_i$ represents the predicted value, $\overline{y}$ is the mean of the actual values, and M is the number of data values being considered. The R^2 score ranges from 0 to 1, where a value of 1 indicates perfect fit of the model to the data, and a value of 0 indicates no fit at all. Therefore, the closer the R^2 score is to 1, the better the model is performing [28].

The *MRE* is the average of the relative errors between the predicted and actual values. A lower *MRE* indicates a better performance of the model on average [29]. It is defined as follows:

$$MRE = \frac{1}{M} \sum_{i=0}^{M} \left| \frac{y_i - \widehat{y}_i}{y_i} \right|. \tag{12}$$

The strategy employed for the hyperparameter tuning process in this work was the Grid Search method [30]. In this method, each hyperparameter is assigned a predefined list of discrete values (called a search space) and every possible combination of these values is tested, with the validation set results being evaluated through (11) and (12). The values that yield the best performance are then selected.

Through the hyperparameter tuning process, the structure of the DNN was defined (see Figure 1). The model that achieved the best performance on the validation set consists of two hidden layers, each with 10 units ($m_1 = m_2 = 10$). The learning rate optimization resulted in a value of 0.01. This model structure and learning rate achieved relatively high R^2 scores (0.9773 for y_1 and 0.9792 for y_2) and low mean relative errors (0.0598 for y_1 and 0.0294 for y_2). Additionally, the model has a relatively low number of trained parameters (the total number of weights and biases), with 262 parameters, offering a good balance between complexity and performance. The optimizer algorithm chosen was the SGD.

A possible application of the developed DNN model is in optical network planning tools, allowing for a quick estimation of network throughput in various scenarios characterized by multiple factors, such as network topology, traffic demands, and physical limitations, to support the design and optimization of optical networks. In the following section, we will see that each network instance can be analysed in time intervals on the order of a few milliseconds.

4. Simulations and discussion

4.1 *Simulation parameters*

To train the DNN model, a dataset of 15,245 networks was used. These networks were generated using the tool described in [12], considering a 2D square plane with side lengths ranging from 1000 km to 5000 km in increments of 1000 km, 4 regions in the plane, a number of nodes varying from 5 to 55, and an average node degree ranging from 2 to approximately 5. The Waxman parameters chosen were $\alpha = \beta = 0.4$. This range of number of nodes was chosen because it covers the values typically found in reference networks [23], while also allowing for a short computation time with the routing algorithm. The constrained routing solution was determined using Algorithm 1 (see Appendix) assuming a full-mesh logical topology and transmission at 64 Gbaud in the C-band, which leads to a limit of 75 optical channels per link.

Table 1 shows the inputs (features), which are the various parameters related to the optical network's physical topology, and the outputs (labels) of the DNN model, presenting their minimum, mean and maximum values. As can be seen, there is a great variability in the network parameters. This is intended to reflect the variability in real-world optical backbone networks, training the model for a wide range of possible networks.

Table 1. Features, labels, and their statistics

Features and labels	Minimum	Mean	Maximum
x_1 : number of nodes	5	33.79	55
x_2 : number of links	5	60.33	138
x_3 : minimum link length (km)	75	109.79	692
x_4 : maximum link length (km)	204	1604.58	5042
x_5 : average link length (km)	119.52	550.30	2260
x_6 : variance of link length (km^2)	968	141,063.71	1,953,209
x_7 : minimum node degree	2	2.04	5
x_8 : maximum node degree	2	6.36	14
x_9 : average node degree	2	3.57	5.14
x_{10} : variance of node degree	0	1.58	6.68
x_{11} : network diameter	1	9.33	27
x_{12}: algebraic connectivity	1.93	217.34	8351.29
y_1 : network throughput (Tb/s)	11.8	514.87	1934.96
y_2 : avg. channel capacity (Gb/s)	211.92	538.79	970

4.2 *Results and discussion*

To evaluate the predictive performance of the DNN model on both randomly generated networks and reference networks, the trained DNN model was used to make predictions on different test sets. The goal of these tests is to assess the model's accuracy in its predictions and compare its prediction times to the computation times of the routing algorithm.

As a first test on the DNN model, a set of 750 random networks was generated under the same conditions as the networks used to train the model, with the routing algorithm also being applied under the same assumptions. The total computation time for the routing algorithm was 7 min and 10 s, while the prediction with the DNN took only 22 ms for the entire set of 750 networks.

The mean relative errors for this test set, as given by (13), are: 6.17% for the network throughput (y_1) and 2.84% for the average channel capacity (y_2) predictions. Figure 2 shows the scatter plot of the relative errors against the number of nodes for both outputs. Each dot represents the relative error (RE) for each individual network in the set, given by

$$RE = \frac{y_i - \widehat{y_i}}{y_i} \tag{13}$$

with y_i being the value determined from the routing solution and $\widehat{y_i}$ the prediction made with the DNN model.

As depicted in Figure 2a, most relative errors for the network throughput are under 10%, with 79.23% of the examples having a relative error below this threshold, 90.81% of examples below 15% and an average relative error of 6.17%. The average channel capacity, illustrated in Figure 2b, generally presents smaller relative errors: 83.36% of examples have a relative error below 5%, 97.60% of the examples fall below 10% and an average relative error of 2.84%.

These results show that, despite providing good performance in both outputs, the model exhibits a superior performance on the average channel capacity. The observed disparity in performance between the two outputs could be indicative of the model's ability to learn and generalize from the training data, suggesting that the features present in the training data could be more predictive for the average channel capacity.

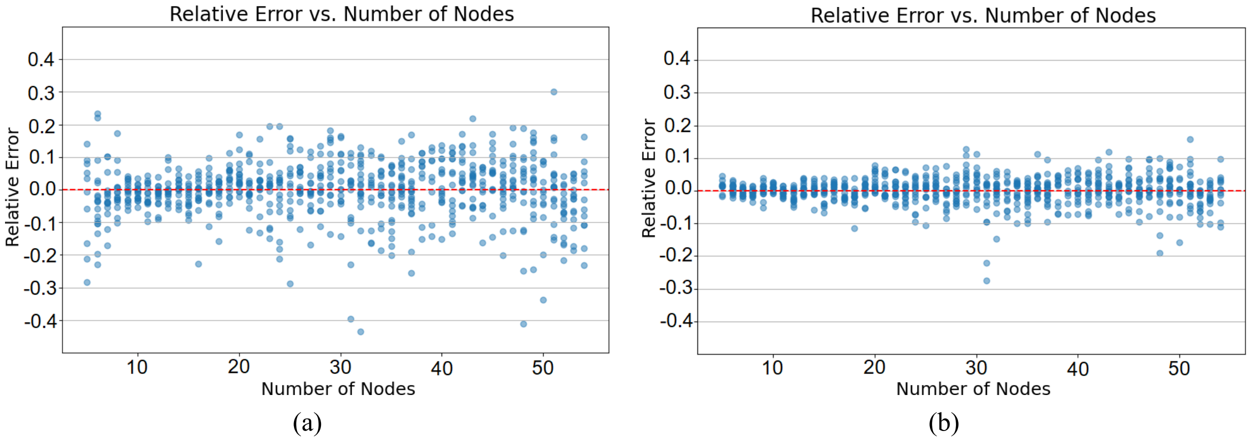

Figure 2. Relative error against number of nodes for both outputs: (**a**) Network throughput; (**b**) Average channel capacity

To test the model's performance on larger networks, a test set with 1440 networks was generated in the same conditions as the previous sets, but now with a number of nodes varying from 5 to 100. The runtime of the routing algorithm was 2 h 53 min, while the predictions with the DNN model took only 63 ms for the entire set. The scatter plot of the relative errors versus the number of nodes for this set of networks is shown in Figure 3.

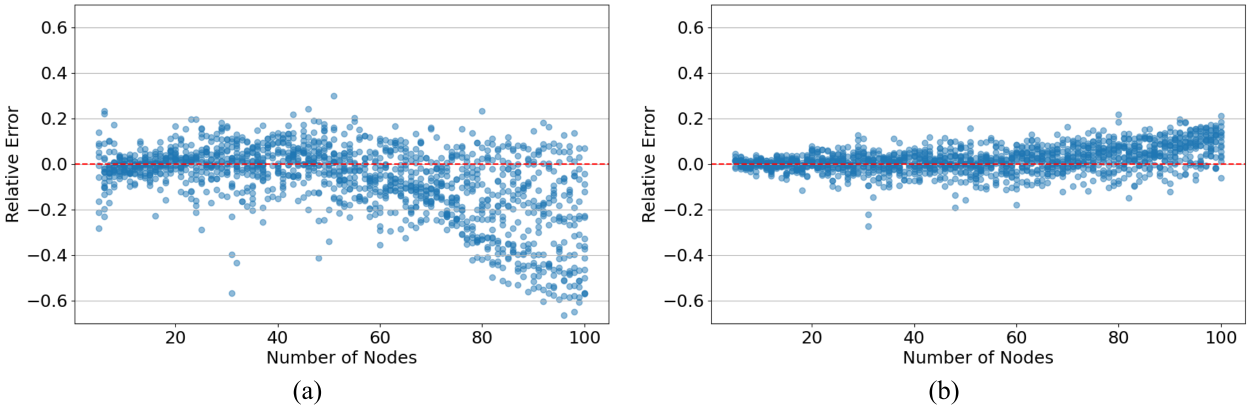

Figure 3. Relative error versus the number of nodes for both outputs (N in [5,100]): (**a**) Network throughput; (**b**) Average channel capacity

From the plots in Figure 3, it is clear that the model's performance in networks with a larger number of nodes is significantly more irregular, becoming worse as the number of nodes increases. This behaviour is expected, as the model has been trained on a specific range of data (networks with a number of nodes ranging from 5 to 55), and extrapolating beyond this range is likely to result in less reliable predictions. The ability of a model to perform well on unseen data (data it was not trained on), is known as out-of-distribution (OOD) generalization [31]. This presents a challenge for conventional supervised learning methods, such as DNNs, which often struggle to handle this effectively, as these models fundamentally assume that the training and test datasets come from the same distribution. Note also that, as the number of nodes increases, parameters such as the number of links and even the outputs themselves will also tend to have values the model was not trained with.

The inaccurate performance is particularly evident in the network throughput (Figure 3a). In this case the predictions tend to stay inside the range of 20% relative error when the number of nodes is lower than 55, but for higher numbers of nodes the relative errors start progressively becoming higher (in absolute value). However, in the case of the average channel capacity (Figure 3b) this effect is much less pronounced, with the relative errors staying inside the 20% range even for larger number of node values. This may indicate that the prediction of this output may be more dependent on other features that do not vary significantly outside their original distribution range or that the output itself does not vary as much with OOD values as the total network throughput does. Nevertheless, it is important to note that, when dealing with OOD generalization, the model's behaviour can be very unpredictable and erratic, and so an evaluation on the model's performance poses a significant challenge [31]. Due to its critical role, OOD generalization remains an active and important topic of research in the field of machine learning [31].

Table 2 shows the DNN model's predictions, and the routing solution values for specific random networks with a varying number of nodes. The time needed to determine the throughput and channel capacity values with the routing algorithm (t_R) is compared with the DNN model's prediction time (t_P).

Table 2. DNN predictions in random network. y_1 : Network throughput; y_2: Avg. channel capacity

N	y_1 (Tb/s)	$\hat{y}_1$ (Tb/s)	$RE\ y_1$(%)	y_2 (Gb/s)	$\hat{y}_2$ (Gb/s)	$RE\ y_2$(%)	t_R (ms)	t_P (ms)
10	48.0	48.67	1.40	533.33	547.08	2.58	190	51.67
20	303.2	303.75	0.18	797.89	801.63	0.47	200	60.32
30	708.0	663.32	6.31	813.79	787.77	3.19	380	24.70
40	292.67	268.64	8.21	333.33	332.92	0.12	430	10.02
50	627.98	530.97	15.45	376.49	342.26	9.09	1970	8.56
60	600.271	718.48	19.69	315.93	331.48	4.92	2490	22.86
70	1080.11	1141.53	5.69	412.26	386.95	6.14	5970	13.59
80	1293.15	1476.62	14.18	334.49	353.83	5.78	21,380	16.89
90	2204.88	1964.20	10.92	463.80	387.0	16.56	38,430	12.31
100	1641.51	1786.44	8.82	314.46	294.91	6.22	56,930	21.60

The DNN model is significantly faster than the routing algorithm, taking only a few milliseconds compared to the routing algorithm's tens of seconds in larger networks. Furthermore, when it comes to the capacity, the same tendencies identified before can be observed here as well: errors are generally smaller for y_2 than for y_1, and larger in networks with a number of nodes in an OOD range. Specifically, in the scenario where $N \in \{10,\ 20, \ldots, 50\}$, the training and testing datasets belong to the same distribution, which is not the case for the scenario where $N \in \{60,\ 70, \ldots, 100\}$. Consequently, in the first scenario, the average relative errors for y_1 and y_2 are 6.31 and 3.09, respectively, values that are quite close to those mentioned early. In contrast, in the second scenario, the corresponding average relative errors increase to 11.86 and 7.92, respectively, highlighting the model's difficulties in generalizing to unseen data. Nevertheless, it is important to note that there is inherent variability in the data that can lead to exceptions to the overall trends, as is the case with the network with 100 nodes, which presents smaller relative errors than some of the networks with fewer nodes.

To assess the performance of the DNN model in real-world networks, four reference networks were considered: COST239 ($N = 11$, $K = 26$, $\bar{l} = 462.6$ km) [32], DTAG ($N = 14$, $K = 23$, $\bar{l} = 236.5$ km) [8], NSFNET ($N = 14$, $K = 21$, $\bar{l} = 1211.3$ km) [32], and UBN ($N = 24$, $K = 43$, $\bar{l} = 993.2$ km) [32], with $\bar{l}$ representing the average link length. These networks are frequently used in networks studies, with the first two typically used to describe medium-size networks and the last two large-size networks. Table 3 compares the throughput and channel capacity values determined with the routing algorithm with the DNN model's predictions, as well as the respective computation times.

Table 3. DNN predictions in reference networks. y_1: Network throughput; y_2: Avg. channel capacity

Network name	y_1 (Tb/s)	$\hat{y}_1$ (Tb/s)	$RE\ y_1$ (%)	y_2 (Gb/s)	$\hat{y}_2$ (Gb/s)	REy_2(%)	t_R (ms)	t_P (ms)
COST239	81.2	80.05	1.42	738.18	730.69	1.02	224.98	10.01
DTAG	147.4	141.64	3.91	809.89	783.61	3.24	174.35	23.01
NSFNET	98.0	96.85	1.17	538.46	543.50	0.94	168.70	22.39
UBN	272.8	262.85	3.65	494.20	472.54	4.39	263.46	22.72

The results show that the DNN model achieves low relative errors in predicting both outputs. This indicates that the model can make accurate predictions in real-world optical backbone networks, further confirming that the topologies of the random networks used in the training of the model describe well their real-world counterparts.

5. Conclusions

This paper has presented a machine learning-based approach to estimate the throughput of optical backbone networks operating in the C-band. To generate the datasets required for training the model, we applied a heuristic based on a constraint routing algorithm, using a full-mesh logical topology. The implemented model was a DNN with 12 inputs, representing parameters related to the physical topology of optical networks, 2 outputs, and 2 hidden layers. The outputs represent the network throughput and the average channel capacity.

The tests performed on the DNN model showed good performance on both artificially generated networks and reference networks. On the set of random networks, the model was able to predict the network throughput with a mean relative error of 6.17%, and the average channel capacity with a mean relative error of 2.84%. However, these results are only valid when the training and testing datasets share the same distribution. If this condition does not hold, the relative errors can increase due to limitations in out-of-distribution (OOD) generalization.

When it comes to the reference networks, a good performance was verified on the four networks tested, with low relative errors verified for both outputs in all cases.

The DNN model also achieved significantly faster performance than the heuristic routing method, with predictions consistently taking only a few tens of milliseconds, whereas the routing algorithm required up to tens of seconds to reach results in larger networks.

Given the DNN model's accurate performance and fast prediction, it could prove to be a valuable resource for developing network planning tools to be used to design optical backbone networks.

Future work will explore approaches to overcome the OOD generalization problem to enhance the DNN capabilities and enable their use across a wider range of applications.

Conflict of interest

Declare conflicts of interest or state “There is no conflict of interest for this study”.

Appendix

This appendix describes the algorithm used to compute the matrix of established paths and the matrix of blocked traffic demands, based on the heuristic approach provided in [12].

Algorithm 1 Constraint Routing Strategy

```
1: **Input:** Weighted graph $G(V,E)$, demand matrix $D=[d_{(s,t)}]$, maximum number of channels per link $N_{ch.max}$.
2: **Output:** Matrix of established paths $\Pi=[\pi_{(s,t)}]$, matrix of blocked paths $B=[b_{(s,t)}]$.
3: for each pair of nodes $(s,t)$ do
4:     Compute the shortest path $(\pi_{(s,t)})$ using Dijkstra's algorithm, with the path length $l(\pi_{(s,t)})$ as the metric;
5: end for
6: for each traffic demand $d_{(s,t)}$ do
7:     Order the demand according to a certain sorting strategy (shortest-first, longest-first, largest-first);
8: end for
9: for each ordered traffic demand $d_{(s,t)}$ do
10:    if there is enough residual capacity (difference between link capacity and load) in all the links of $\pi_{(s,t)}$ AND an
       enough available number of channels then
11:        Route the demand $d_{(s,t)}$ over the path $\pi_{(s,t)}$ and update the load (number of demands) on all the links traversed by
           the path;
12:        Assign a wavelength to the channel corresponding to the demand using the first-fit strategy;
13:    else
14:        Block the traffic demand $d_{(s,t)}$;
15:    end if
16:    Remove from $G(V,E)$ all links with a residual capacity of zero;
17:    for each pair of nodes $(s,t)$ do
18:        Search for new shortest paths using Dijkstra's algorithm;
19:    end for
20:    Go to step 7;
21: end for
22: **Return:** $\Pi$ and $B$.
```

References

[1] P. J. Winzer and D. T. Neilson, "From scaling disparities to integrated parallelism: A decathlon for a decade," *J. Lightw. Technol.*, vol. 35, no. 5, pp. 1099–1115, 2017, https://doi.org/10.1109/JLT.2017.2662082.

[2] L. Nadal, M. Ali, F. J. Vílchez, J. M. Fàbrega, and M. S. Moreolo, "The multiband over spatial division multiplexing sliceable transceiver for future optical networks," *Future Internet*, vol. 15, no. 12, p. 381, 2023, https://doi.org/10.3390/fi15120381.

[3] A. Ferrari, E. Virgillito, and V. Curri, "Band-division vs. space-division multiplexing: a network performance statistical assessment," *J. Lightw. Technol.*, vol. 38, no. 5, pp. 1041–1049, 2020, https://doi.org/10.1109/JLT.2020.2970484.

[4] C. E. Shannon, "A Mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, pp. 379–423, 1948, https://doi.org/10.1002/j.1538-7305.1948.tb01338.x.

[5] R.-J. Essiambre, G. Kramer, P. J. Winzer, G. J. Foschini, and B. Goebel, "Capacity limits of optical fiber networks," *J. Lightw. Technol.*, vol. 28, no. 4, pp. 662–701, 2010, https://doi.org/10.1109/JLT.2009.2039464.

[6] G. Bosco, P. Poggiolini, A. Carena, V. Curri, and F. Forghieri, "Analytical results on channel capacity in uncompensated optical links with coherent detection," *Opt. Express*, vol. 19, no. 26, pp. B438–B449, 2011, https://doi.org/10.1364/OE.19.00B440.

[7] P. Bayvel, R. Maher, T. Xu, G. Liga, N. A. Shevchenko, D. Lavery, et al., "Maximizing the optical network capacity," *Philos. Trans. R. Soc. A*, vol. 374, p. 2014044, 2016, https://doi.org/10.1098/rsta.2014.0440.

[8] D. J. Ives, P. Bayvel, and S. J. Savory, "Routing, modulation, spectrum and launch power assignment to maximize the traffic throughput of a nonlinear optical mesh network," *Photon. Netw. Commun.*, vol. 29, pp. 244–256, 2015, https://doi.org/10.1007/s11107-015-0488-0.

[9] R. J. Vincent, D. J. Ives, and S. J. Savory, "Scalable capacity estimation for nonlinear elastic all-optical core networks," *J. Lightw. Technol.*, vol. 37, no. 21, pp. 5380–5391, 2019, https://doi.org/10.1109/JLT.2019.2942710.

[10] R. Matzner, D. Semrau, R. Luo, G. Zervas, and P. Bayvel, “Making intelligent topology design choices: understanding structural and physical property performance implications in optical networks,” *J. Opt. Commun. Netw.*, vol. 13, no. 8, pp. D53–D67, 2021, https://doi.org/10.1364/JOCN.423490.

[11] K. Higashimori, T. Tanaka, F. Inuzuka, T. Ohara, T. Inoue, “Key physical topology features for optical backbone networks via a multilayer correlation,” *J. Opt. Commun. Netw.*, vol. 15, no. 5, pp. B23–B32, 2023, https://doi.org/10.1364/JOCN.479866.

[12] A. Freitas and J. Pires, “Investigating the impact of topology and physical impairments on the capacity of an optical backbone network,” *Photonics*, vol. 11, no. 4, p. 342, 2024, https://doi.org/10.3390/photonics11040342.

[13] A. Souza, N. Costa, J. Pedro, and J. Pires, “Comparison of fast quality of transmission estimation methods for C+ L + S optical systems,” *J. Opt. Commun. Netw.*, vol. 15, no. 11, pp. F1–F13, 2023, https://doi.org/10.1364/JOCN.486898.

[14] F. Musumeci, C. Rottondi, A. Nag, I. Macaluso, D. Zibar, M. Ruffini, et al., “An Overview on application of machine learning techniques in optical networks,” *IEEE Commun. Surv. Tut.*, vol. 21, no. 2, pp. 1383–1408, 2019, https://doi.org/10.1109/COMST.2018.2880039.

[15] Y. Y. Bay and K. A. Yearick, “Machine learning vs deep learning: the generalization problem,” *arXiv*, 2024, https://doi.org/10.48550/arXiv.2403.01621.

[16] Y. Cheng, Y. Guo, M. Cao, Y. Jiang, W. Ren, G. Ren, “Few-mode fiber design for multiple-input-multiple-output-less mode division multiplexing by machine learning,” *J. Opt. Soc. Am. B*, vol. 29, pp. 2421–2428, 2022, https://doi.org/10.1364/JOSAB.462459.

[17] R. Morais and J. Pedro, “Machine learning models for estimating quality of transmission in DWDM networks,” *J. Opt. Commun. Netw.*, vol. 10, no. 10, pp. D84–D99, 2018, https://doi.org/10.1364/JOCN.10.000D84.

[18] K. Hornik, M. Stinchcombe, and H. White, “Multilayer feedforward networks are universal approximators,” *Neural Netw.*, vol. 2, no. 5, pp. 359–366, 1989, https://doi.org/10.1016/0893-6080(89)90020-8.

[19] Y. Zhang, Y. Song, Y. Shi, J. Li, C. Zhang, Y. Tang, et al., “Deep Neural Network-enabled Fast and Large-Scale QoT Estimation for Dynamic C+L-Band Mesh Networks,” in *Proc. Optical Fiber Commun. Conf.*, San Diego, CA, USA, Mar. 7–9, 2023.

[20] T. Tanimura, T. Hoshida, J. C. Rasmussen, M. Suzuki, H. Morikawa, “OSNR Monitoring by Deep Neural Networks Trained with Asynchronously Sampled Data,” in *Proc. 21st OptoElectronics Commun. Conf./Int. Conf. Photonics Switching*, Niigata, Japan, Jul. 3–7, 2016, paper TuB3-5.

[21] J. Yu, B. Cheng, C. Hang, Y. Hu, S. Liu, Y. Wang, et al., “A Deep Learning Based RSA Strategy for Elastic Optical Networks,” in *Proc. 18th Int. Conf. Optical Commun. Netw.*, Huangshan, China, Aug. 5–8, 2019, https://doi.org/10.1109/ICOCN.2019.8934862.

[22] B. Châtelain, M. P. Bélanger, C. Tremblay, F. Gagnon, and D. V. Plant, “Topological Wavelength Usage Estimation in Transparent Wide Area Networks,” *J. Opt. Commun. Netw.*, vol. 1, no. 1, pp. 196–203, 2009, https://doi.org/10.1364/JOCN.1.000196.

[23] C. Pavan, R. M. Morais, J. R. F. Rocha, and A. N. Pinto, “Generating Realistic Optical Transport Network Topologies,” *J. Opt. Commun. Netw.*, vol. 2, no. 1, pp. 80–90, 2010, https://doi.org/10.1364/JOCN.2.000080.

[24] F. Kuipers, T. Korkmaz, M. Krunz, and P. V. Mieghem, “An overview of constraint-based path selection algorithms for QoS routing,” *IEEE Commun. Mag.*, vol. 40, pp. 50–55, 2003, https://doi.org/10.1109/MCOM.2002.1106159.

[25] A. Souza, B. Correia, N. Costa, J. Pedro, and J. Pires, “Accurate and scalable quality of transmission estimation for wideband optical systems,” in *Proc. IEEE Int. W. Comput.-Aided Modeling Anal. Design Commun. Links Netw.*, Porto, Portugal, Oct. 25–27, 2021.

[26] S. Chugh, S. Ghosh, A. Gulistan, and B. Rahman, “Machine Learning Regression Approach to the Nanophotonic Waveguide Analyses,” *J. Lightw. Technol.*, vol. 37, no. 24, pp. 6080–6089, 2019, https://doi.org/10.1109/JLT.2019.2946572.

[27] Y. Lecun, “A theoretical framework for back-propagation,” in *Proc. 1988 Conn. Models Summer Sch.*, Pittsburgh, PA, USA, Jun. 17–26, 1988, pp. 21–28.

[28] Coefficient of determination, r-squared. Accessed: Jul. 7, 2024. [Online]. Available: https://www.ncl.ac.uk/webtemplate/ask-assets/external/maths-resources/statistics/regression-and-correlation/coefficient-of-determination-r-squared.html.

[29] MRE—Mean Relative Error. Accessed: Jul. 7, 2024. [Online]. Available: https://permetrics.readthedocs.io/en/latest/pages/regression/MRE.html.

[30] J. Brownlee, "Hyperparameter optimization with random search and grid search." Accessed: Jul. 7, 2024. [Online]. https://machinelearningmastery.com/hyperparameter-optimization-with-random-search-and-grid-search/.

[31] Z. Shen, J. Liu, Y. He, X. Zhang, R. Xu, H. Yu, et al., "Towards out-of-distribution generalization: a survey," *arXiv*, 2021, https://doi.org/10.48550/arXiv.2108.13624.

[32] J. J. O. Pires, "On the Capacity of Optical Backbone Networks," *Network*, vol. 4, no. 1, pp. 114–132, 2024, https://doi.org/10.3390/network4010006.